

APPLYING GENERATIVE AI IN INTRODUCTORY PROGRAMMING: A SURVEY OF INFORMATION TECHNOLOGY STUDENTS AT SELECTED REGIONAL UNIVERSITIES

Phan Thi Gam^{*1}, Nguyen Quoc Dung²

* Corresponding author:
Email: gam.phanthi@htu.edu.vn

¹ Email: dung.nguyenquoc@htu.edu.vn

^{1,2} Ha Tinh University
Cam Binh commune, Ha Tinh province,
Vietnam

Received: 16/10/2025

Revised: 02/12/2025

Accepted: 25/12/2025

Published: 20/01/2026

Abstract: Generative Artificial Intelligence (GenAI) is increasingly transforming higher education by enabling innovation in teaching and learning while posing challenges related to accuracy, academic integrity, and dependence. Although international research has highlighted the potential of GenAI in programming education, empirical studies in Vietnam - especially at regional universities - remain limited. This study examines the current use of GenAI tools among Information Technology students in introductory programming courses, exploring their perceptions, challenges, and opportunities for integration. Survey results reveal that most students have already utilized GenAI tools, such as ChatGPT and GitHub Copilot, primarily for basic tasks like syntax explanation and code generation. In contrast, advanced uses such as code verification or debugging are still limited. Based on these findings, the study proposes pedagogical solutions for integrating GenAI into programming instruction to enhance students' self-directed learning, foster the safe and ethical use of technology, and develop problem-solving and critical thinking skills.

Keywords: Artificial Intelligence, GenAI application, basic programming, ChatGPT, GitHub Copilot.

NGHIÊN CỨU THỰC TRẠNG VÀ ĐỀ XUẤT GIẢI PHÁP ỨNG DỤNG TRÍ TUỆ NHÂN TẠO SINH TRONG DẠY HỌC LẬP TRÌNH TẠI MỘT SỐ TRƯỜNG ĐẠI HỌC

Phan Thị Gấm^{*1}, Nguyễn Quốc Dũng²

* Tác giả liên hệ:
Email: gam.phanthi@htu.edu.vn

² Email: dung.nguyenquoc@htu.edu.vn

^{1,2} Trường Đại học Hà Tĩnh
Xã Cẩm Bình, tỉnh Hà Tĩnh, Việt Nam

Nhận bài: 16/10/2025

Chỉnh sửa xong: 02/12/2025

Chấp nhận đăng: 25/12/2025

Xuất bản: 20/01/2026

Tóm tắt: Trí tuệ nhân tạo, đặc biệt là trí tuệ nhân tạo tạo sinh (Generative AI - GenAI), đang tác động mạnh mẽ đến giáo dục đại học. Công nghệ này vừa mở ra cơ hội đổi mới phương pháp dạy học vừa đặt ra thách thức về tính chính xác, đạo đức học thuật và nguy cơ phụ thuộc. Các nghiên cứu quốc tế đã chỉ ra tiềm năng của GenAI trong hỗ trợ học tập và giảng dạy lập trình. Tuy nhiên, ở Việt Nam, nhất là tại các trường đại học địa phương, các nghiên cứu thực nghiệm để phân tích và đánh giá tình trạng này còn hạn chế và chưa có định hướng ứng dụng rõ ràng. Nghiên cứu này nhằm đánh giá thực trạng và đề xuất giải pháp ứng dụng GenAI trong học lập trình cơ bản của sinh viên ngành Công nghệ thông tin. Kết quả khảo sát ở một số trường đại học địa phương cho thấy phần lớn sinh viên đã tiếp cận GenAI nhưng mới dừng ở mức khai thác cơ bản, chỉ đáp ứng các yêu cầu phổ thông và chưa thực hiện được các nhiệm vụ nâng cao như kiểm chứng thông tin hay cải tiến mã nguồn. Trên cơ sở đó, nghiên cứu đề xuất các giải pháp tích hợp GenAI vào hoạt động học tập lập trình, nhằm nâng cao hiệu quả tự học, phát triển năng lực sử dụng công nghệ một cách sáng tạo, an toàn và có trách nhiệm.

Từ khóa: Trí tuệ nhân tạo, ứng dụng GenAI, lập trình cơ bản, ChatGPT, GitHub Copilot.

1. Đặt vấn đề

Trong bối cảnh chuyển đổi số diễn ra mạnh mẽ trên phạm vi toàn cầu, trí tuệ nhân tạo (Artificial Intelligence - AI) ngày càng khẳng định vai trò quan

trọng trong việc đổi mới phương thức quản lý, giảng dạy và nâng cao chất lượng giáo dục. Trong đó, trí tuệ nhân tạo tạo sinh (Generative AI - GenAI) nổi

lên như một công nghệ tiên tiến có tiềm năng ứng dụng rộng rãi trong giáo dục đại học. GenAI không chỉ hỗ trợ giảng viên trong việc thiết kế và biên soạn học liệu, đổi mới phương pháp dạy học mà còn mở ra nhiều cơ hội trong việc hỗ trợ tự học, cá nhân hóa quá trình học tập và nghiên cứu của người học (Gleeson, 2024; Lau và Guo, 2023).

Trong đào tạo công nghệ thông tin, năng lực lập trình là năng lực cốt lõi, giúp sinh viên phát triển tư duy logic, khả năng giải quyết vấn đề và sáng tạo công nghệ (Bubnic và các cộng sự, 2024). Tuy nhiên, thực tế dạy học cho thấy, nhiều sinh viên gặp khó khăn khi bắt đầu học lập trình do lượng kiến thức lớn, cấu trúc lệnh phức tạp và nhiều khái niệm trừu tượng, dẫn đến tâm lý e ngại làm giảm động lực học tập (Medeiros và cộng sự, 2018; Banaw và cộng sự, 2024). Trước tình hình đó, sự xuất hiện của các công cụ GenAI như ChatGPT, GitHub Copilot, Gemini hay Codex mang đến cách tiếp cận mới giúp sinh viên viết mã, gỡ lỗi, mô phỏng thuật toán và hiểu khái niệm lập trình dễ dàng hơn (Hossain Shihab, 2025; McCulloh và cộng sự, 2025).

Các nghiên cứu tổng quan gần đây ghi nhận sự gia tăng mạnh mẽ về số lượng và phạm vi nghiên cứu ứng dụng GenAI trong giáo dục Tin học. Agbo và cộng sự (2025) tổng hợp 126 bài báo khoa học và cho thấy số lượng công trình đã tăng gấp 5 lần trong giai đoạn từ năm 2022 đến năm 2024 so với trước đây, phản ánh rõ sự quan tâm ngày càng lớn của các nhà khoa học đến chủ đề này. Garcia (2025) cũng khẳng định việc tích hợp GenAI trong dạy học lập trình là xu thế tất yếu và mang lại cơ hội đổi mới phương pháp sư phạm. Tuy nhiên, các tổng quan này chỉ ra nhiều khoảng trống nghiên cứu như thiếu bằng chứng thực nghiệm dài hạn, chưa thống nhất trong cách đo lường, thiếu công cụ đánh giá năng lực lập trình và nhu cầu phát triển khung hướng dẫn để tích hợp GenAI trong lĩnh vực CNTT (Garcia, 2025; Rubio-Manzano và cộng sự, 2025).

Nhiều nghiên cứu thực nghiệm đã cung cấp bằng chứng rõ ràng về tác động của GenAI đối với việc học lập trình. Rahe và Maalej (2025) chỉ ra nguy cơ phụ thuộc vào công nghệ khi sinh viên có xu hướng dựa vào GenAI để viết toàn bộ lời giải. Ngược lại, Dickey và cộng sự (2025) cho rằng, ứng dụng GenAI bằng cách tổ chức môi trường học tập có hướng dẫn (AI-Lab) giúp sinh viên hình thành năng lực tự đánh giá và phát triển kỹ năng sử dụng GenAI hiệu quả. Thực nghiệm của Hossain Shihab (2025) với GitHub Copilot cũng ghi nhận GenAI hỗ trợ hoàn thành

nhệm vụ nhanh và hiệu quả nhưng dễ dẫn đến hiểu sai khái niệm nếu thiếu định hướng sư phạm. Các thực nghiệm cũng chỉ ra GenAI có thể nâng cao động lực và sự tự tin, song không cải thiện đáng kể kết quả học tập nếu thiếu hướng dẫn sư phạm (Sun và cộng sự, 2024; Claudio và cộng sự, 2025). Ngoài ra, nhiều nghiên cứu nhấn mạnh yêu cầu bồi dưỡng kỹ năng sử dụng và viết câu lệnh (Prompt) để khai thác GenAI hiệu quả hơn trong các nhiệm vụ học tập (Alves và Cipriano, 2025).

Bên cạnh đó, một số nghiên cứu tập trung vào thử nghiệm các mô hình sư phạm tích hợp GenAI, trong đó mô hình học đảo ngược được áp dụng phổ biến. Thực nghiệm với ChatGPT cho thấy mô hình học bằng cách “dạy lại công cụ” giúp cải thiện đáng kể mức độ tương tác và kết quả học tập (Huesca và cộng sự, 2024). Tương tự, Liang (2025) áp dụng mô hình học đảo ngược trong học phần Python cơ bản cũng ghi nhận GenAI thúc đẩy sáng tạo và tính chủ động của người học. Những bằng chứng này cho thấy hiệu quả của GenAI phụ thuộc nhiều vào mô hình sư phạm, cần cân bằng giữa hỗ trợ công nghệ và phát triển năng lực tư duy lập trình cốt lõi.

Tại Việt Nam, nghiên cứu về ứng dụng AI trong giáo dục đại học mới ở giai đoạn khởi đầu. Phần lớn công trình tập trung vào việc nhận diện tác động và tiềm năng ứng dụng, song còn thiếu nghiên cứu thực nghiệm chuyên sâu, đặc biệt trong đào tạo lập trình. Kết quả nghiên cứu cho thấy, GenAI có thể mở rộng phạm vi học tập, cá nhân hóa nội dung và thúc đẩy tự học nhưng cũng tiềm ẩn nguy cơ lệ thuộc và vi phạm đạo đức học thuật (Bùi Văn Tài và Nguyễn Minh Tuấn, 2024). Một số nghiên cứu về hệ thống cá nhân hóa và đánh giá tự động cho thấy GenAI nâng cao mức độ tham gia và hiệu quả học tập, song gặp thách thức về hạ tầng và năng lực triển khai, đặc biệt tại các trường địa phương (Nguyen và Phan, 2024). Nghiên cứu của Phan và cộng sự (2025) chỉ ra rằng, việc tích hợp AI vào giáo dục Việt Nam gặp rào cản về an ninh dữ liệu, hạ tầng công nghệ, năng lực giảng viên... và để áp dụng rộng rãi cần giải quyết triệt để các vấn đề này nhằm đảm bảo ứng dụng an toàn và hiệu quả.

Các báo cáo tại AI4Edu 2024 do Câu lạc bộ các Khoa - Trường - Viện Công nghệ thông tin - Truyền thông Việt Nam (FISU) tổ chức cũng bước đầu ghi nhận hiệu quả tích cực của trợ lý AI trong dạy lập trình, song nhấn mạnh rằng cần nghiên cứu đánh giá một cách hệ thống các tác động của GenAI và có các khuyến nghị sư phạm cần thiết để ứng dụng công nghệ này một cách bền vững.

Đặc biệt, tại các trường đại học địa phương, việc ứng dụng GenAI trong dạy học lập trình còn hạn chế, thiếu quy định và hướng dẫn cụ thể, dẫn đến việc triển khai không đồng bộ và chưa tận dụng tối đa tiềm năng của công nghệ. Thực trạng này đặt ra nhu cầu cấp thiết phải khảo sát, đánh giá tác động và nhận thức của sinh viên về GenAI trong quá trình học lập trình, từ đó làm cơ sở để xây dựng chiến lược ứng dụng phù hợp.

Xuất phát từ thực trạng trên, nghiên cứu này tập trung tìm hiểu việc ứng dụng GenAI trong học lập trình cơ bản của sinh viên tại một số trường đại học thông qua các câu hỏi nghiên cứu sau:

RQ1: Sinh viên lựa chọn các công cụ GenAI nào và sử dụng chúng cho những mục đích gì trong quá trình học lập trình cơ bản?

RQ2: Sinh viên hiện đang sở hữu những năng lực nào trong việc khai thác GenAI để hỗ trợ học lập trình?

RQ3: Sinh viên nhận thức như thế nào về các tác động tích cực và hạn chế của GenAI đối với quá trình học lập trình?

RQ4: Sinh viên gặp khó khăn gì khi sử dụng GenAI trong học lập trình và cần những giải pháp nào để nâng cao hiệu quả ứng dụng GenAI trong bối cảnh này?

Kết quả nghiên cứu cung cấp cơ sở khoa học để nhận diện những vấn đề còn tồn tại và đề xuất các giải pháp sư phạm nhằm nâng cao hiệu quả ứng dụng GenAI trong giảng dạy lập trình ở bậc đại học. Đồng thời, nghiên cứu cũng đóng góp bằng chứng thực nghiệm cho việc tích hợp GenAI trong dạy học lập trình tại Việt Nam.

2. Phương pháp nghiên cứu

Nghiên cứu sử dụng kết hợp phương pháp định lượng và định tính nhằm đánh giá thực trạng sử dụng GenAI trong hoạt động học lập trình của sinh viên ngành Công nghệ thông tin.

Khảo sát được tiến hành đối với sinh viên năm thứ nhất và năm thứ hai tại hai trường đại học địa phương khu vực Bắc Trung Bộ, vào cuối năm học 2024 - 2025, trong bối cảnh chưa có quy định chính thức về việc ứng dụng GenAI trong đào tạo. Tổng cộng thu về 85 phiếu trả lời hợp lệ, phản ánh sự đa dạng về giới tính, trình độ lập trình và mức độ trải nghiệm với GenAI.

Phiếu khảo sát được thiết kế trực tuyến trên nền tảng Google Form gồm năm nhóm câu hỏi. Cấu trúc và nội dung câu hỏi được xây dựng dựa trên khung

Lý thuyết hợp nhất về chấp nhận và sử dụng công nghệ (UTAUT) (Momani, 2020), các tiếp cận về năng lực lập trình trong đào tạo khoa học máy tính (ACM, IEEE-CS, AACL, 2023) và các nghiên cứu liên quan đến ứng dụng GenAI trong giáo dục lập trình (Alves và Cipriano, 2024; Hashmi và cộng sự, 2024; Scholl và Kiesler, 2024; Rahe và cộng sự, 2025). Phiếu được rà soát và hoàn thiện thông qua tham vấn chuyên gia trong lĩnh vực công nghệ giáo dục. Nội dung phiếu khảo sát bao gồm:

1) Thông tin chung: Thu thập đặc điểm giới tính, trình độ, kinh nghiệm lập trình trước đại học, mức độ tiếp xúc với GenAI và các quy định về việc sử dụng GenAI tại các cơ sở giáo dục.

2) Công cụ và mục đích sử dụng GenAI: Xác định các công cụ được sử dụng, tần suất và mục đích ứng dụng trong học lập trình cơ bản.

3) Kỹ năng sử dụng GenAI: Khảo sát khả năng mô tả yêu cầu (prompt), đánh giá và chỉnh sửa kết quả, đặc biệt đối với mã nguồn do GenAI sinh ra.

4) Nhận thức về tác động của GenAI: Đo lường đánh giá của sinh viên về hiệu quả và hạn chế của GenAI trong các nhiệm vụ lập trình và quá trình học tập.

5) Nhận định và đề xuất: Thu thập ý kiến mở về việc ứng dụng GenAI trong học lập trình và các đề xuất giải pháp để cải thiện.

Dữ liệu định lượng được xử lý bằng phần mềm SPSS, sử dụng các phép thống kê mô tả và phân tích bảng chéo để xác định đặc điểm mẫu và xu hướng sử dụng GenAI của sinh viên.

Dữ liệu định tính, thu thập từ các câu hỏi mở, được phân loại theo nội dung và thống kê tần suất xuất hiện, kết hợp với kết quả định lượng để làm rõ các nhận định về khó khăn, yếu tố ảnh hưởng, giải pháp cải thiện và định hướng ứng dụng GenAI trong học lập trình. Quá trình phân tích có sử dụng GPT-4 như một công cụ hỗ trợ. Tuy nhiên, toàn bộ đánh giá cuối cùng được thực hiện bởi nhóm nghiên cứu nhằm đảm bảo tính khách quan.

Kết quả phân tích cung cấp cơ sở để nhận diện thách thức và đề xuất giải pháp ứng dụng GenAI phù hợp với đặc thù đào tạo lập trình tại các trường đại học địa phương.

3. Kết quả nghiên cứu

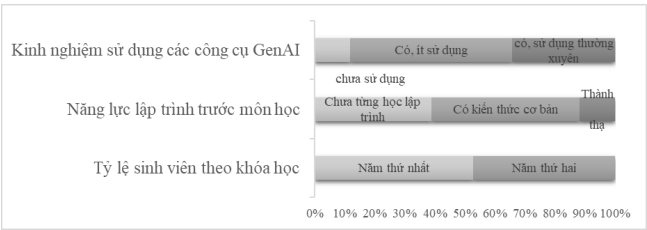
3.1. Thực trạng ứng dụng trí tuệ nhân tạo tạo sinh trong dạy học lập trình

3.1.1. Đặc điểm mẫu khảo sát

Kết quả khảo sát thu được 85 phiếu trả lời hợp lệ,

trong đó sinh viên năm thứ nhất chiếm 52.9% (n = 45) và sinh viên năm thứ hai chiếm 47.1% (n = 40).

Về năng lực lập trình trước khi học lập trình cơ bản ở bậc Đại học, có 38.8% sinh viên chưa tiếp cận lập trình, 61.2% sinh viên đã biết lập trình, trong đó 11.8% tự đánh giá khả năng lập trình ở mức khá thành thạo. Kết quả này cho thấy, phần lớn sinh viên tham gia khảo sát chưa có nền tảng lập trình vững chắc, phù hợp với mục tiêu nghiên cứu tập trung vào người học mới làm quen với lập trình.



Hình 1: Đặc điểm của mẫu khảo sát

Về mức độ tiếp cận các công cụ GenAI, chỉ 11.8% sinh viên chưa từng sử dụng, trong khi 54.1% sử dụng các tính năng cơ bản, 34.1% có khả năng sử dụng thành thạo cho nhiều mục đích khác nhau. Như vậy, hơn 88% sinh viên đã tiếp cận GenAI, trong đó trên 34% sử dụng thường xuyên. Kết quả này phản ánh xu hướng nhanh chóng tiếp nhận các công cụ hỗ trợ học tập dựa trên trí tuệ nhân tạo, ngay cả khi phần lớn chưa có nền tảng lập trình vững chắc.

3.1.2. Các công cụ GenAI và mục đích sử dụng GenAI trong học lập trình

Kết quả khảo sát về lựa chọn các công cụ GenAI hỗ trợ học lập trình (xem Bảng 1) cho thấy, ChatGPT là công cụ được sử dụng phổ biến nhất ở tất cả các nhóm với tỉ lệ trung bình đạt 89.3%. Trong đó, nhóm sinh viên chưa từng học lập trình sử dụng nhiều nhất (93.75%), tiếp theo là nhóm có kiến thức lập trình cơ

bản (88.1%) và nhóm khá thành thạo (80%). Điều này cho thấy, ChatGPT là công cụ có tính tiếp cận cao, đặc biệt phù hợp với sinh viên mới làm quen với lập trình. Nhận định này cũng phù hợp với các nghiên cứu quốc tế, trong đó ChatGPT được ghi nhận là công cụ GenAI được sử dụng phổ biến nhất trong giáo dục Tin học (Agbo và cộng sự, 2025).

Đối với GitHub Copilot, tỉ lệ sử dụng tăng dần theo trình độ lập trình: 9.1% ở nhóm chưa từng lập trình, 23.8% ở nhóm có kiến thức cơ bản và 30% ở nhóm thành thạo. Điều này phản ánh rằng, Copilot phù hợp hơn với sinh viên đã có nền tảng lập trình, phù hợp với tổng hợp của Agbo và cộng sự (2025), cho thấy Copilot thường được khai thác hiệu quả hơn ở người học có kinh nghiệm do yêu cầu hiểu mã và làm việc trong môi trường IDE.

Hai công cụ GenAI khác là Claude và Gemini được sử dụng phổ biến hơn ở nhóm sinh viên có năng lực lập trình thành thạo, với tỉ lệ sử dụng lần lượt là 30% đối với Claude và 60% đối với Gemini. Ngoài ra, một số sinh viên đề xuất thêm các công cụ GenAI khác như DeepSeek và Grok để hỗ trợ học lập trình.

Về mục đích sử dụng GenAI, kết quả khảo sát (xem Bảng 2) cho thấy, 72.9% sinh viên sử dụng GenAI để tìm hiểu khái niệm, cú pháp và cấu trúc lập trình, chủ yếu ở những người mới làm quen với lập trình. Hơn một nửa sinh viên sử dụng GenAI để gợi ý cách giải bài tập hoặc tìm thuật toán (56.5%) hoặc sửa lỗi chương trình (52.9%), phản ánh vai trò trực tiếp của GenAI trong thực hành viết mã.

Các nhiệm vụ nâng cao như tối ưu hóa, cải tiến mã nguồn (50.6%), viết chú thích hoặc đặc tả chương trình (35.3%) và trả lời câu hỏi ôn tập, kiểm tra (36.5%) được ít lựa chọn hơn. Chỉ 30.6% sinh viên sử dụng GenAI để sinh đoạn mã mẫu cho thấy ưu tiên

Bảng 1: Tỉ lệ sử dụng công cụ theo năng lực lập trình

Năng lực lập trình trước môn học	Tỉ lệ trung bình sử dụng công cụ				n
	ChatGPT	GitHub Copilot	Claude	Gemini	
Chưa từng học lập trình	0.938	0.091	0.121	0.273	33
Có kiến thức cơ bản	0.881	0.238	0.119	0.214	42
Khá thành thạo	0.800	0.300	0.300	0.600	10
Tổng	0.893	0.188	0.141	0.282	85

Mean = tỉ lệ trung bình sử dụng công cụ trong nhóm (1 = có sử dụng, 0 = không sử dụng)

Bảng 2: Mục đích sử dụng GenAI hỗ trợ học lập trình

STT	Mục đích sử dụng GenAI	Số lượng (n)	Tỉ lệ (%)
1	Tìm hiểu khái niệm, cú pháp, cấu trúc lệnh	62	72.9%
2	Gợi ý cách giải bài tập hoặc thuật toán	48	56.5%
3	Sinh đoạn mã mẫu	26	30.6%
4	Tìm và sửa lỗi chương trình	45	52.9%
5	Tối ưu, cải tiến mã nguồn	43	50.6%
6	Viết chú thích hoặc đặc tả chương trình	30	35.3%
7	Trả lời câu hỏi ôn tập, kiểm tra	31	36.5%

Bảng 3: Kỹ năng sử dụng GenAI của sinh viên theo năm học

STT	Kỹ năng sử dụng GenAI	Năm 1	Năm 2	TB chung
1	Biết cách đặt câu hỏi (prompt) rõ ràng và cụ thể.	0.525	0.825	0.675
2	Biết đánh giá và kiểm chứng kết quả GenAI tạo ra.	0.425	0.750	0.588
3	Hiểu và phân tích được mã nguồn do GenAI tạo ra.	0.400	0.725	0.563
4	Có khả năng chỉnh sửa và cải tiến mã nguồn do GenAI sinh ra.	0.425	0.675	0.550
5	Biết kết hợp với tài liệu và nguồn học tập khác để nâng cao chất lượng học tập.	0.333	0.550	0.442

tìm hiểu, chỉnh sửa và hoàn thiện chương trình thay vì sao chép trực tiếp.

So với nghiên cứu của Scholl và Kiesler (2024) thực hiện trên 250 sinh viên quốc tế (tỉ lệ 60% cho tìm hiểu khái niệm/cú pháp, 45% sửa lỗi chương trình), kết quả khảo sát này cao hơn 10 - 15%. Sự chênh lệch cho thấy đối tượng khảo sát hiện tại chủ yếu là sinh viên mới học lập trình, có trình độ đầu vào phù hợp với các trường địa phương khi kiến thức nền tảng và kỹ năng lập trình còn hạn chế trong khi mẫu quốc tế gồm cả sinh viên năm cuối và người học có kinh nghiệm. Kết quả khảo sát này cho thấy, sinh viên mới phụ thuộc nhiều hơn vào GenAI để củng cố kiến thức nền tảng, còn người học có kinh nghiệm thường dùng công cụ cho các hoạt động nâng cao như kiểm thử, tối ưu và phân tích hiệu suất.

3.1.3. Kỹ năng sử dụng GenAI trong lập trình

Kết quả khảo sát kỹ năng sử dụng GenAI trong học lập trình (xem Bảng 3) cho thấy, 67.% sinh viên ở các trường địa phương tương tác với GenAI ở mức khá, khi 67.5% biết viết prompt rõ ràng, cụ thể. Tuy

nhiên, kỹ năng phân biện và kiểm chứng còn hạn chế, chỉ 58.75% đánh giá được kết quả GenAI tạo ra, phản ánh mức phụ thuộc cao vào công cụ thay vì vào quy trình tư duy và kiểm thử. Hơn một nửa số sinh viên có khả năng hiểu, phân tích mã nguồn và biết chỉnh sửa, cải tiến các đoạn mã do GenAI đề xuất. Tỉ lệ sinh viên biết kết hợp GenAI với tài liệu và nguồn tham khảo khác để nâng cao chất lượng học lập trình chiếm 44.17%, thấp hơn các kỹ năng khác. Kết quả cũng chỉ ra rằng, các kỹ năng sử dụng GenAI của sinh viên thay đổi đáng kể giữa sinh viên năm thứ nhất (52.5%) và năm thứ hai (82.5%), phản ánh đây là kỹ năng nền tảng được cải thiện dần trong quá trình học tập.

Về việc xử lý kết quả do GenAI cung cấp dựa trên việc phân tích định tính dữ liệu khảo sát cho thấy đa số sinh viên có xu hướng thận trọng và kiểm chứng thông tin trước khi sử dụng. Tuy nhiên, một bộ phận sinh viên vẫn tin tưởng và sử dụng trực tiếp kết quả từ công cụ, tiềm ẩn nguy cơ tiếp nhận thông tin hoặc mã nguồn không chính xác.

3.1.4. Nhận thức về tác động của GenAI đối với việc học lập trình

Kết quả khảo sát (xem Bảng 4) cho thấy, nhìn chung sinh viên đánh giá GenAI hỗ trợ hiệu quả trong đa số các nhiệm vụ lập trình, đặc biệt ở các hoạt động tìm hiểu lí thuyết và viết mã mới. Các giá trị trung bình đều nằm trong khoảng 4.00 - 4.15, tương ứng mức “Hiệu quả” đến “Rất hiệu quả”, qua đó phản ánh xu hướng sử dụng GenAI như một trợ lí học tập.

Khi đối chiếu với nghiên cứu của Haindl và Weinberger (2024), mức đánh giá của sinh viên trong nghiên cứu này cao hơn rõ rệt ở các hoạt động như “viết mã mới” (4.13 so với 3.56) và “tối ưu hóa thuật toán, đoạn mã” (4.12 so với 3.98), trong khi các nhiệm vụ như “tìm hiểu lí thuyết” và “viết tài liệu” lại cho thấy mức độ tương đồng. Điều này cho thấy, GenAI trong bối cảnh khảo sát không chỉ được sử dụng để hỗ trợ tiếp cận kiến thức mà còn được khai thác ở các khâu chuyên sâu hơn của quy trình phát triển mã nguồn. Kết quả này góp phần mở rộng các ghi của các nghiên cứu trước, nhấn mạnh vai trò của GenAI ở mức cung cấp khái niệm nền tảng (Agbo và cộng sự, 2025).

Phân tích theo nhóm năm học cho thấy sự khác biệt nhất quán giữa sinh viên năm nhất và năm hai ở tất cả các tiêu chí, trong đó nhóm năm hai luôn có mức trung bình cao hơn. Ở hai tiêu chí được đánh giá cao, mức chênh lệch lần lượt là 4.24 so với 4.05 (tìm hiểu lí thuyết) và 4.29 so với 3.95 (viết đoạn mã mới). Xu hướng này tiếp tục thể hiện ở các nhiệm vụ đòi hỏi kĩ năng cao hơn, như tối ưu hóa thuật toán (4.29 so với 3.85) và thiết kế giải pháp lập trình (4.27 so với

3.95). Những khác biệt này gợi ý rằng, kinh nghiệm và nền tảng kiến thức đóng vai trò quan trọng trong việc khai thác GenAI cho các hoạt động đòi hỏi tư duy thuật toán, đồng thời góp phần giải thích độ lệch chuẩn tương đối lớn trong thống kê mô tả.

Các nhiệm vụ liên quan đến quy trình viết mã cũng được đánh giá tích cực. Hoạt động “viết tài liệu hoặc chú thích mã nguồn” (Mean = 4.14; SD = 0.86) đều có mức đánh giá cao ở cả hai nhóm, phù hợp với nhận định rằng, GenAI hỗ trợ tốt các thao tác hoàn thiện sản phẩm lập trình (Rahe và cộng sự, 2025). Đối với các nhiệm vụ mang tính tư duy thuật toán sâu hơn như “Tối ưu hóa thuật toán” (Mean = 4.08) và “Thiết kế giải pháp lập trình” (Mean = 4.12), kết quả thu được tương đương hoặc nhỉnh hơn so với Haindl và Weinberger (2024) (Mean = 3.98). Hoạt động “Sửa lỗi chương trình (debug)” (Mean = 4.00; SD = 0.94) có độ phân tán lớn nhất, phản ánh sự khác biệt đáng kể trong khả năng vận dụng GenAI vào gỡ lỗi. Điều này phù hợp với đặc thù của nhiệm vụ, vốn đòi hỏi người học phải mô tả chính xác bối cảnh lỗi và đánh giá tính hợp lí của gợi ý do GenAI đưa ra - một kĩ năng mà sinh viên năm hai thể hiện tốt hơn (Mean = 4.13 so với 3.85). Kết quả này cũng tương đồng với ghi nhận của Rahe và cộng sự (2025) rằng, hiệu quả sử dụng GenAI trong hoạt động gỡ lỗi phụ thuộc mạnh vào năng lực phân tích của người học.

Về hạn chế của GenAI, Bảng 5 cho thấy, sinh viên nhận thức khá rõ về rủi ro khi sử dụng công cụ này (Mean = 3.61, SD ≈ 0.80). Hạn chế được ghi nhận nhiều nhất là “Khó đưa ra giải pháp toàn diện cho bài toán lớn hoặc đòi hỏi tư duy logic/chuyên sâu” (Mean = 3.85, SD = 0.84), tiếp đến là “Thông tin/mã

Bảng 4: Thống kê mô tả về các mục đích sử dụng GenAI trong học lập

STT	Mục đích sử dụng GenAI	Năm thứ 1 (n=45)		Năm thứ 2 (n=40)		Trung bình	
		Mean	SD	Mean	SD	Mean	SD
1	Tìm hiểu lí thuyết, khái niệm lập trình.	4.05	0.78	4.24	0.91	4.15	0.85
2	Viết đoạn mã mới.	3.95	0.81	4.29	0.76	4.13	0.80
3	Sửa lỗi chương trình (Debug).	3.85	0.95	4.13	0.92	4.00	0.94
4	Tối ưu hóa thuật toán, đoạn mã.	3.85	0.98	4.29	0.79	4.08	0.90
5	Thiết kế thuật toán, đưa ra giải pháp lập trình.	3.95	0.93	4.27	0.91	4.12	0.93
6	Viết tài liệu hoặc chú thích cho mã nguồn.	4.03	0.80	4.24	0.91	4.14	0.86

(Mean - Giá trị trung bình, SD - Độ lệch chuẩn.)

Bảng 5: Đánh giá về các hạn chế của GenAI trong lập trình

STT	Hạn chế	Mean	SD
1	Thông tin/mã nguồn tạo ra sai hoặc lỗi thời.	3.70	0.78
2	Giải pháp không đáp ứng đầy đủ yêu cầu bài toán.	3.55	0.80
3	Mã nguồn thiếu chú thích, không giải thích rõ cơ chế hoạt động.	3.45	0.79
4	Khó đưa ra giải pháp toàn diện cho bài toán lớn hoặc đòi hỏi tư duy logic/chuyên sâu.	3.85	0.84
5	Thiếu trích dẫn nguồn, khó xác minh tính chính xác.	3.50	0.77

nguồn sai hoặc lỗi thời” (Mean = 3.70, SD = 0.78) và “Giải pháp không đáp ứng đầy đủ yêu cầu” (Mean = 3.55, SD = 0.80). Các vấn đề như “Mã nguồn thiếu chú thích, không giải thích rõ cơ chế hoạt động” (Mean = 3.45, SD = 0.79) và “Thiếu trích dẫn nguồn, khó xác minh tính chính xác” (Mean = 3.50, SD = 0.77) phản ánh điểm yếu về minh bạch và khả năng hỗ trợ học tập sâu.

Từ các kết quả trên, có thể thấy GenAI đã trở thành công cụ hỗ trợ toàn diện, từ việc tiếp thu kiến thức lí thuyết, thực hành lập trình, đến phân tích và cải thiện mã nguồn. Tuy vậy, để phát huy tối đa hiệu quả ở các nhiệm vụ nâng cao như tối ưu thuật toán hay gỡ lỗi phức tạp, sinh viên cần rèn luyện các kĩ năng viết câu lệnh yêu cầu (Prompt engineering), xác minh tính đúng đắn của mã nguồn và tối ưu giải pháp, phù hợp với phân tích trước đó về kĩ năng sử dụng GenAI

3.2. Một số vấn đề khi sử dụng GenAI trong dạy học lập trình

Sự xuất hiện của các công cụ GenAI đã làm thay đổi cách học lập trình. Kết quả khảo sát cho thấy quá trình khai thác GenAI của sinh viên Công nghệ thông tin tồn tại nhiều hạn chế, phản ánh cả sự chưa đầy đủ về kĩ năng công nghệ lẫn thách thức sự phạm trong dạy học lập trình.

Thứ nhất, nguy cơ lệ thuộc vào GenAI và hạn chế phát triển tư duy lập trình. Kazemitabaar và cộng sự (2023) khẳng định rằng, hiệu quả của GenAI phụ thuộc đáng kể vào năng lực nền tảng của người học. Kết quả khảo sát của chúng tôi củng cố nhận định này, phần lớn sinh viên chưa có kiến thức lập trình nhưng lại tiếp cận GenAI rất sớm. Kĩ năng viết prompt của sinh viên năm nhất (Mean = 0.525) thấp hơn đáng kể so với sinh viên năm hai (Mean = 0.825). Xu hướng tương tự xuất hiện ở hầu hết các kĩ năng khác, cho thấy phần lớn sinh viên năm nhất chưa thực sự hiểu

cách mô tả mục tiêu, bối cảnh bài toán hoặc kiểm tra kết quả, khiến GenAI trở thành nguồn cung cấp lời giải nhanh, thay thế các hoạt động tư duy lập trình cần thiết.

Dữ liệu khảo sát cũng phản ánh xu hướng sinh viên năm nhất sử dụng GenAI để nhận lời giải trực tiếp mà không phân tích bài toán, thử nghiệm hay gỡ lỗi. Nhóm năm hai dù có năng lực tốt hơn nhưng mức thành thạo trung bình vẫn chỉ đạt dưới 0.725 ở các kĩ năng cốt lõi. Việc bỏ qua các bước phân tích, lập luận và tự xây dựng thuật toán làm suy giảm tư duy logic và năng lực giải quyết vấn đề, vốn là nền tảng của học lập trình (Kazemitabaar và cộng sự, 2023; Lau & Guo, 2023; Prather và cộng sự, 2024).

Đáng chú ý, ở bối cảnh Việt Nam, hầu hết các trường đại học chưa ban hành quy định chính thức hoặc triển khai chương trình đào tạo về sử dụng GenAI. Vì vậy, sinh viên tiếp cận và sử dụng công cụ này theo hướng tự phát, đặc biệt ở nhóm năm nhất vốn thiếu kiến thức nền và kĩ năng học thuật. Sự kết hợp giữa việc sử dụng tự phát và nền tảng lập trình yếu làm gia tăng mức độ lệ thuộc vào GenAI, có thể rõ rệt hơn so với bối cảnh quốc tế.

Thứ hai, nguy cơ tiếp nhận thông tin hoặc lời giải sai lệch. Mặc dù GenAI hỗ trợ hiệu quả trong việc gợi ý mã và giải thích khái niệm, nhưng mức độ chính xác không được đảm bảo tuyệt đối. Khảo sát cho thấy chỉ 58.75% sinh viên biết cách đánh giá và kiểm chứng kết quả GenAI tạo ra. Trong khi đó, các hạn chế về chất lượng đầu ra của GenAI được ghi nhận ở mức cao. Những sai lệch này dễ bị bỏ qua do văn phong trôi chảy của GenAI, khiến sinh viên dễ bị “ảo tưởng về độ chính xác” (Prather và cộng sự, 2024). Mức độ phổ biến của lỗi được sinh viên ghi nhận trong khảo sát này cao hơn so với nhiều nghiên cứu quốc tế, phản ánh sự khác biệt về kĩ năng số và kinh nghiệm lập trình trong bối cảnh nghiên cứu.

Thứ ba, nguy cơ hạn chế kỹ năng phân tích, gỡ lỗi và kiểm chứng thông tin. Sử dụng GenAI làm giảm nhu cầu thực hành các kỹ năng phân tích và gỡ lỗi, những thành phần cốt lõi để phát triển năng lực lập trình (Momani, 2020). Dữ liệu khảo sát cho thấy phần lớn sinh viên chưa có khả năng kiểm chứng và đánh giá đầu ra của GenAI và bỏ qua việc đối chiếu phản hồi với tài liệu chính thống hoặc tự phát hiện sai sót trong mã nguồn. Tương tự, kỹ năng phân tích mã nguồn mà GenAI tạo ra cũng chỉ ở mức trung bình (0.5625). Khi sinh viên không đủ năng lực đánh giá chất lượng nội dung, họ dễ chấp nhận kết quả một cách thụ động dẫn tới suy giảm kỹ năng nhận diện lỗi, kiểm thử, năng lực tự giải quyết vấn đề và gia tăng rủi ro tiếp nhận thông tin sai lệch, đặc biệt đối với các nhiệm vụ lập trình phức tạp, yêu cầu kỹ thuật cao (Lau & Guo, 2023; Prather và cộng sự, 2024).

Thứ tư, nguy cơ vi phạm đạo đức học thuật và gian lận trong đánh giá. GenAI có thể bị lạm dụng trong bài tập hoặc kiểm tra. Với khả năng sinh mã nhanh và cú pháp đúng, các công cụ như ChatGPT hay GitHub Copilot có thể giúp sinh viên hoàn thành yêu cầu học tập dễ dàng, từ đó tăng nguy cơ sao chép hoặc gian lận. Khảo sát cho thấy sinh viên cũng quan ngại về vấn đề bản quyền mã nguồn (Mean = 3.50; SD = 0.77). Điều này đặt ra thách thức kép: 1) Kết quả đánh giá không phản ánh đúng năng lực thực sự, làm suy giảm tính trung thực trong đào tạo; 2) Việc sử dụng từ GenAI mà không ghi chú có thể dẫn đến vi phạm bản quyền. Nhận định này đồng nhất với nghiên cứu của Lund và các cộng sự (2025), khi chỉ ra rủi ro học thuật liên quan đến tính minh bạch và quyền sở hữu trí tuệ trong môi trường giáo dục có sự tham gia của GenAI.

Nhìn chung, việc ứng dụng GenAI trong học lập trình tiềm ẩn nhiều thách thức về chất lượng học tập, năng lực thực hành và đạo đức nghề nghiệp. Những vấn đề này không chỉ phản ánh nhận thức của sinh viên mà còn đặt ra yêu cầu xây dựng các giải pháp sư phạm phù hợp, được trình bày ở mục 3.3.

3.3. Một số giải pháp nâng cao hiệu quả sử dụng GenAI trong dạy học lập trình

Để khai thác GenAI hiệu quả, đồng thời hạn chế các rủi ro về lệ thuộc, sai lệch thông tin và vi phạm đạo đức học thuật, cần triển khai các giải pháp toàn diện, kết hợp nâng cao năng lực người học, thiết kế bài giảng phù hợp và xây dựng chính sách quản lý rõ ràng.

Thứ nhất, đào tạo kỹ năng khai thác GenAI. Hiệu quả của GenAI phụ thuộc trực tiếp vào năng lực khai

thác công cụ của sinh viên. Các cơ sở giáo dục cần xây dựng chương trình nâng cao năng lực số và kỹ năng sử dụng AI trong học tập. Nội dung đào tạo bao gồm: Kỹ thuật đặt câu hỏi rõ ràng, khả năng phân tích và kiểm thử mã do GenAI gợi ý cũng như rèn luyện kỹ năng hiệu chỉnh và tối ưu hóa mã nguồn.

Bên cạnh các hướng dẫn kỹ thuật, chương trình cần nhấn mạnh rằng, GenAI là công cụ hỗ trợ, không phải nguồn giải pháp sẵn có. Kết quả do GenAI cung cấp không đảm bảo chính xác tuyệt đối và cần được kiểm chứng với các nguồn tài liệu khác. Đồng thời, tư duy thuật toán, khả năng giải quyết vấn đề và tư duy phản biện cần được rèn luyện thường xuyên qua các bài tập phân tích, giúp sinh viên nhận diện giải pháp đúng nhưng chưa tối ưu và giải pháp sai nhưng hợp lý, từ đó học cách lựa chọn và cải tiến phương án. Để tăng cường thực hành, cần tổ chức các workshop và bài tập hướng dẫn theo nhóm hoặc cá nhân, với các kịch bản khai thác GenAI trong những tình huống lập trình thực tế, giúp sinh viên vừa phát triển kỹ năng vừa duy trì tư duy sáng tạo.

Thứ hai, xây dựng quy định và khung hướng dẫn sử dụng GenAI. Cần thiết lập quy định và khung hướng dẫn chi tiết, xác định rõ nội dung, phạm vi và cách thức áp dụng GenAI trong học tập. Khung hướng dẫn nên phân loại các hoạt động có thể sử dụng GenAI, từ hỗ trợ tìm hiểu khái niệm, gợi ý mã nguồn, giải thích thuật toán đến phát hiện và sửa lỗi, đồng thời xác định mức độ sử dụng (tham khảo, hỗ trợ hay thay thế). Khung hướng dẫn cũng cần quy định nguyên tắc minh bạch về trích dẫn nguồn và phạm vi sử dụng, đồng thời đề xuất các biện pháp xử lý khi phát hiện vi phạm nhằm hình thành ý thức chủ động và trung thực học thuật cho sinh viên.

Thứ ba, cập nhật chương trình đào tạo và đổi mới phương pháp dạy học tích hợp GenAI. Chương trình học cần kết hợp kiến thức lập trình truyền thống với các hoạt động khai thác GenAI, nhằm tối ưu hóa phát triển năng lực. Nội dung có thể được tổ chức thành hai phần: 1) Nội dung thực hiện hoàn toàn dựa trên năng lực cá nhân, không sử dụng GenAI, nhằm phát triển tư duy thuật toán, khả năng phân tích và giải quyết vấn đề độc lập; 2) Nội dung có hỗ trợ GenAI, sinh viên phải mô tả chi tiết quá trình sử dụng, điều chỉnh và cải tiến giải pháp để lựa chọn phương án tối ưu.

Giảng viên đóng vai trò định hướng và mô hình hóa cách khai thác GenAI có trách nhiệm, tạo điều kiện cho sinh viên phát triển tư duy phản biện và khai thác công cụ hiệu quả. Các hoạt động học tập

được thiết kế liên kết giữa học cá nhân, thực hành nhóm và thảo luận chéo, giúp sinh viên so sánh, phân tích và cải tiến giải pháp, đồng thời nâng cao khả năng tự học, tự đánh giá và hợp tác.

Hệ thống đánh giá nên đa dạng hóa để phản ánh đúng năng lực thực chất: bài tập lập trình thủ công kiểm tra năng lực nền tảng; bài tập sử dụng GenAI nhưng yêu cầu trình bày chi tiết quá trình áp dụng; báo cáo hoặc thảo luận cá nhân nhằm phân tích và cải tiến giải pháp. Đồng thời, sử dụng các công cụ để phát hiện sao chép mã nguồn tạo sinh, đảm bảo tính trung thực và công bằng.

Triển khai đồng bộ các giải pháp trên sẽ giúp ứng dụng GenAI trong dạy học lập trình không chỉ đóng vai trò hỗ trợ kỹ thuật mà còn trở thành động lực thúc đẩy tư duy phản biện, năng lực phân tích, khả năng sáng tạo và tinh thần tự học của sinh viên.

4. Kết luận

Nghiên cứu này cho thấy, sinh viên ngành Công nghệ thông tin tiếp cận các công cụ GenAI, đặc biệt là ChatGPT và GitHub Copilot ở mức độ cao, sử dụng với nhiều mục đích khác nhau. Kết quả khảo sát và phân tích thực nghiệm chỉ ra rằng, GenAI hỗ trợ hiệu quả cả việc tiếp thu lý thuyết lẫn thực hành lập trình, nhất là trong việc giải thích mã nguồn và hướng dẫn thuật toán. Đồng thời, nghiên cứu cũng làm rõ những hạn chế: Sinh viên có nguy cơ lệ thuộc vào kết quả GenAI, giải pháp do GenAI tạo ra đôi khi sai lệch hoặc lỗi thời và tiềm ẩn nguy cơ làm giảm tư duy phản biện độc lập hoặc vi phạm đạo đức học thuật. Những phát hiện này nhấn mạnh nhu cầu triển khai GenAI trong dạy học lập trình theo hướng có kiểm soát và định hướng sự phạm, gắn liền với rèn luyện tư duy phản biện, kỹ năng phân tích và

kiểm chứng kết quả. Trên cơ sở đó, nghiên cứu đề xuất ba nhóm giải pháp trọng tâm: 1) Phát triển kỹ năng sử dụng GenAI an toàn, hiệu quả và nâng cao nhận thức về bản quyền, đạo đức học thuật; 2) Xây dựng khung hướng dẫn sử dụng GenAI, xác định mục đích, mức độ và nguyên tắc minh bạch; 3) Thiết kế nội dung học tập kết hợp giữa nhiệm vụ thực hiện độc lập và tham khảo GenAI có giải trình, đổi mới phương pháp dạy học, giáo viên giữ vai trò hướng dẫn, giám sát sinh viên sử dụng GenAI một cách có chủ đích, đồng thời đổi mới hệ thống đánh giá, kết hợp đo lường năng lực tư duy logic, khả năng phân tích và giải trình khi ứng dụng GenAI.

Kết quả nghiên cứu có ý nghĩa cả về lý luận và thực tiễn: Về lý luận, cung cấp bằng chứng thực nghiệm về tác động của GenAI trong bối cảnh học lập trình cơ bản; Về thực tiễn, định hướng việc tích hợp GenAI vào giảng dạy một cách có kiểm soát, giúp sinh viên phát triển năng lực lập trình thực chất, tư duy phản biện và khả năng sáng tạo. Tuy nhiên, nghiên cứu vẫn còn hạn chế về quy mô mẫu, phạm vi khảo sát chỉ tập trung ở một cơ sở đào tạo và chưa đi sâu vào đo lường tác động dài hạn. Các nghiên cứu tiếp theo cần được mở rộng theo hướng thực nghiệm lâu dài, có sự tham gia của nhiều trường đại học, đồng thời phân tích sự khác biệt giữa các nhóm sinh viên theo kinh nghiệm, mức độ ứng dụng GenAI, ngôn ngữ lập trình sử dụng và độ phức tạp của bài toán.

Có thể khẳng định rằng, GenAI giữ vai trò quan trọng trong quá trình học tập lập trình, song giá trị thực sự chỉ được phát huy khi được tích hợp có định hướng, gắn liền với giải trình, kiểm chứng và tuân thủ chuẩn mực học thuật.

Tài liệu tham khảo

- Agbo, F. J., Olivia, C., Oguiibe, G., Sanusi, I. T., Sani, G. (2025). Computing education using generative artificial intelligence tools: A systematic literature review. *Computers and Education Open*. <https://doi.org/10.1016/j.caeo.2025.100266>
- Altarawneh, H. (2023). *ChatGpt Impact on Student Educational Performance: A Conceptual Analysis*. <https://doi.org/10.4108/eetel.4574>
- Alves, P. and Cipriano, B. P. (2025). "Give Me the Code": Log Analysis of First-Year CS Students'. *Proceedings of the 17th International Conference on Computer Supported Education*, pp. 198-207. <https://doi.org/10.5220/0013194400003932>
- Association for Computing Machinery (ACM), IEEE Computer Society (IEEE-CS) & Association for the Advancement of Artificial Intelligence (AAAI). (2023). *Computer Science Curricula 2023 (CS2023 curricular guidelines)*. <https://www.acm.org/articles/bull>
- Bubnic, B., Mernik, M., Kosar, T. (2024). Exploring the Predictive Potential of Complex Problem-Solving in Computing Education: A Case Study in the Introductory Programming Course. *Mathematics*. <https://doi.org/10.3390/math12111655>
- Bùi Trọng Tài, Nguyễn Minh Tuấn. (2024). Nghiên cứu ảnh hưởng của trí tuệ nhân tạo trong giáo dục tới hoạt động học tập của sinh viên. *VJE Tạp chí Giáo dục*, tr.6-11. <https://scholar.dlu.edu.vn/thuvienso/bitstream/DLU123456789/240703/1/CVv216S10V242024006.pdf>

- Claudio, C., Rafael M., Daniel C., Enrique U. (2025). Generative Artificial Intelligence in Computer Programming: Does It Enhance Learning, Motivation, and the Learning Environment? *IEEE Access*, pp.40438 - 40455. <https://doi.org/10.1109/ACCESS.2025.3532883>
- Dickey, E., Bejarano, A., Kuperus, R., & Fagundes, B. (2025). Evaluating the AI-Lab Intervention: Impact on Student Perception and Use of Generative AI in Early Undergraduate Computer Science Courses. *ArXiv: Computers and Society*. <https://doi.org/10.48550/arXiv.2505.00100>
- Garcia, M. B. (2025). Teaching and learning computer programming using ChatGPT: A rapid review of literature amid the rise of generative AI technologies. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-025-13452-5>
- Haindl, P. & Weinberger, G. (2024). Students' experiences of using ChatGPT in an undergraduate programming course. Department of Computer Science and Security. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2024.3380909>
- Hossain Shihab, M. I. (2025). The Effects of GitHub Copilot on Computing Students' Programming Effectiveness, Efficiency, and Processes in Brownfield Programming. *arXiv*. <https://doi.org/10.48550/arXiv.2506.10051>
- Huesca, G., Martínez-Treviño, Y., Molina-Espinosa, J. M., Sanromán-Calleros, A. R., Martínez-Román, R., Cendejas-Castro, E. A., & Bustos. (2024). Effectiveness of Using ChatGPT as a Tool to Strengthen Benefits of the Flipped Learning Strategy. *Education Sciences*. <https://doi.org/10.3390/educsci14060660>
- Kazemitabaar, M., Chow, J., Ma, C. K. T., Ericson, B. J., Weintrop, D. & Grossman, T. (2023). Studying the effect of AI code generators on supporting novice learners in introductory programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. AMC. <https://doi.org/10.1145/3544548.3580919>
- Lau, S. & Guo, P. J. (2023). From "ban it till we understand it" to "resistance is futile": How university programming instructors plan to adapt as more students use AI code generation and explanation tools such as ChatGPT and GitHub Copilot. *Proceedings of the 2023 ACM Conference on International Computing Education Research V.1*. <https://doi.org/10.1145/3568813.3600138>
- Liang, J. (2025). Artificial Intelligence Empowered Teaching Reform and Exploration of Python Programming Course. *Open Access Library Journal*. <https://doi.org/10.4236/oalib.1112859>
- Lund, B. D., Lee, T. H., Mannuru, N. R. & Arutla, N. (2025). AI and academic integrity: Exploring student perceptions and implications for higher education. *Journal of Academic Ethics*. <https://doi.org/10.1007/s10805-025-09613-3>
- McCulloh, I., Rodriguez, P., Kumar, S., Gupta, M., Sharma, V. R., Johnson, B., Johnson, A. N. (2025). Generative AI in Computer Science Education: Accelerating Python Learning with ChatGPT. *arXiv*. <https://doi.org/10.48550/arXiv.2505.20329>
- Medeiros, R. P., Ramalho, G. L. & Falcão, T. P. (2018). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE Transactions on Education*. <https://doi.org/10.1109/TE.2018.2864133>
- Momani, A. M. (2020). Momani, A. M. (2020). The Unified Theory of Acceptance and Use of Technology: A new approach in technology acceptance. *International Journal of Sociotechnology and Knowledge Development*, 12(3), pp.79-91. <https://doi.org/10.4018/IJSKD.2020070105>
- Nguyen, N. Q. T. & Phan, T., T. (2024). The Role of AI in Improving Student Learning Outcomes: Evidence in Vietnam. *International Journal of Multidisciplinary research and Analysis*, pp.2806-2810. <https://doi.org/10.47191/ijmra/v7-i06-48>
- Prather, J., Reeves, B., Leinonen, J., MacNeil, S., Randrianasolo, A. S., Becker, B., Kimmel, B., Wright, J. & Briggs, B. (2024). The widening gap: The benefits and harms of generative AI for novice programmers. In *Proceedings of the ACM Conference on International Computing Education Research*. ACM. <https://doi.org/10.48550/arXiv.2405.17739>
- Phan, T. A., Cung, T. L. C., Pham, T. T., Pham, C. B. S., & Pham, T. B. S. (2025). Application of Artificial Intelligence (AI) in Online Education in Vietnam: Opportunities and Challenges. *Research Review International Journal of Multidiscip*. <https://doi.org/10.31305/rrijm.2025.v10.n7.035>
- Rahe C., Maalej W. (2025). How Do Programming Students Use Generative AI? *Proceedings of the ACM on Software Engineering*, pp. 978-1000. <https://doi.org/10.1145/3715762>
- Scholl, A. & Kiesler, N. (2024). How Novice Programmers Use and Experience ChatGPT when Solving Programming Exercises in an Introductory Course. *IEEE ASEE Frontiers in Education Conference*. <https://doi.org/10.48550/arXiv.2407.20792>
- Sun, D., Boudouaia, A., Zhu, C. & Li, Y. (2024). Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study. *International Journal of Educational Technology in Higher Education*. <https://doi.org/10.1186/s41239-024-00446-5>